MARKOV MODELS OF MULTI-ECHELON, REPAIRABLE-ITEM INVENTORY SYSTEMS

by

Donald Gross
Leonidas C. Kioussis
Douglas R. Miller
Richard M. Soland

GWU/IMSE/Serial T-490/84
8 June 1984

The George Washington University
School of Engineering and Applied Science
Department of Operations Research
Institute for Management Science and Engineering

Grant ECS-8200837-01
National Science Foundation
and
Contract N00014-83-K-0217
Office of Naval Research

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Department of Operations Research
Institute for Management Science and Engineering

Abstract
of
GWU/IMSE/Serial T-490/84
8 June 1984

MARKOV MODELS OF MULTI-ECHELON, REPAIRABLE-ITEM INVENTORY SYSTEMS

by

Donald Gross
Leonidas C. Kioussis
Douglas R. Miller
Richard M. Soland

Exact models of finite end-item population, finite repair capacity repairable-item systems are developed using Markov process analyses for both transient and steady state environments. Unlike most currently used multi-echelon models, the infinite population, infinite repair capacity restrictions are removed. Exponential failure and repair times are assumed and the system is modeled as a closed Markovian queuing network.

In the transient case, the finite set of differential equations, and in the steady-state case, the finite set of difference equations, are solved by numerical techniques. The adequacy of these techniques for yielding solutions to *practical* systems is also discussed.

Research Supported by

Grant ECS-8200837-01
National Science Foundation
and
Contract N00014-83-K-0217
Office of Naval Research

THE GEORGE WASHINGTON UNIVERSITY
School of Engineering and Applied Science
Institute for Management Science and Engineering

MARKOV MODELS OF MULTI-ECHELON, REPAIRABLE-ITEM INVENTORY SYSTEMS

by

Donald Gross
Douglas R. Miller

1. Introduction

Consider a typical multi-echelon repairable-item inventory system as shown schematically in Figure 1. Shown there is a *two* location (bases), *two* level of supply (spares at bases and depot), *two* level of repair (base and depot) system which we shall denote as a (2,2,2) system. The nodes BUi (i = 1,2) represent operating and spare units (we consider for now only a single item such as a final assembly or a key component) at base i , BRi (i = 1,2) represent the repair facility at base i , DU represents depot spares, and DR the depot repair facility.

Our goal is to develop exact mathematical models for such finite calling population (finite number of items), finite repair capacity, repairable item provisioning systems in both time-varying and steady-state environments. Specifically, we wish to find the state probability vector (the probability distribution for the system being in its various

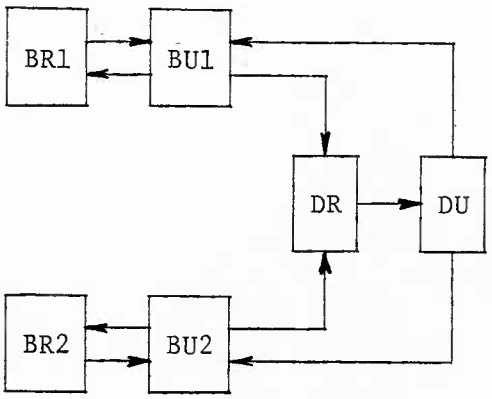


Figure 1. Multi-echelon, repairable item system.

possible states) which will allow us to then calculate measures of performance such as availability (the probability that at least some desirable, prespecified number of components is operational). Ultimately, these models will be used to yield the optimal combination of spares and repair channels at each location in the system.

Assuming times to component failure and component repair times to be exponentially distributed random variables, we have a continuous time Markov process (CTMP). The process is driven by a rate matrix $Q = \{q_{ij}\}$, where q_{ij} is the "rate" of going from state i to state j; that is, letting $X(t)$ represent the system state at time t,

$$q_{ij} = \lim_{\Delta t \to 0} \left[\frac{\Pr\{X(t+\Delta t) = j \mid X(t) = i\}}{\Delta t} \right] , \quad i \neq j ;$$

$$q_{ii} = - \sum_{\substack{j \\ (i \neq j)}} q_{ij} .$$

For example, suppose the $(2,2,2)$ system pictured in Figure 1 is in a state (call it i) for which the depot spares pool is not

empty (at least one spare is on hand at the depot). Suppose we consider the event: a component fails at base 1. Describing this state i by the vector $(n_{BU1}, n_{BR1}, n_{BU2}, n_{BR2}, n_{DU}, n_{DR})$, where n_k denotes the number of components at node k in the "network," this event takes the system to a state j, namely, $(n_{BU1}, n_{BR1}, n_{BU2}, n_{BR2}, n_{DU}-1, n_{DR}+1)$, at the rate $q_{ij} = \lambda\alpha_1 n_{BU1}$, where $1/\lambda$ is the mean time to failure (MTTF) of a component and α_1 is the probability (or percentage) of failed items requiring depot repair.

If we denote the state probability (row) vector at time t by $\underset{\sim}{\pi}(t) = (\pi_1(t), \pi_2(t), \ldots, \pi_S(t))$, that is, the ith element, $\pi_i(t)$, is the probability of the system being in state i at time t (there is a finite number of states [call this number S] even though this number can be quite large), then we must solve the finite set of first-order, linear differential (Kolmogorov) equations

$$\underset{\sim}{\pi}'(t) = \underset{\sim}{\pi}(t)Q \ . \tag{1}$$

For steady-state solutions, we are required to solve the finite set of linear algebraic steady state equations,

$$\underset{\sim}{0} = \underset{\sim}{\pi}Q \ , \tag{2}$$

where $\underset{\sim}{\pi} = (\pi_1, \pi_2, \ldots, \pi_3)$ is the steady-state probability vector and $\underset{\sim}{0}$ is a row vector of all zeroes. In both steady-state and transient cases we have the further condition that the probabilities sum to one, namely,

$$1 = \underset{\sim}{\pi}(t)\underset{\sim}{e} = \underset{\sim}{\pi}\underset{\sim}{e} \ ,$$

where $\underset{\sim}{e}$ is a column vector, with all components equal to 1.

2. Transient Environment

We are often interested in what happens to such systems in a time-varying environment. For example, a sudden increase in effort (say a peacetime to wartime shift) may cause a sudden decrease in MTTF. In such situations, it is necessary to have $\underset{\sim}{\pi}(t)$, and we must solve the finite set of first order linear differential equations given in (1). Except for very small systems (one or two states) analytical techniques such as Laplace transforms are intractable. Since we have a finite set of equations, numerical methods can be employed. Numerical integration schemes such as Runge-Kutta or predictor-corrector methods are possibilities. We choose a different approach, however, which is referred to by some as *randomization*, and has been shown to be more efficient for these kinds of problems [see Arsham, Balana, and Gross (1983) or Grassmann (1977a)]. For details on this technique, which can be derived by a probabilistic argument when viewing the CTMP in a certain way, see Grassmann (1977a and b) or Gross and Miller (1984a and b).

The computational formulas are as follows. Consider a discrete time Markov chain (DTMC) with single-step transition probability matrix

$$P = Q/\Lambda + I \ ,$$

where

$$\Lambda = \max_i |q_{ii}| \ ,$$

that is, Λ is the maximum of the absolute values on the diagonal of the Q matrix. Since a diagonal element of Q is the negative of the sum of the other elements in the row (rows of the Q matrix sum to zero), Λ is actually the absolute value of the minimum (largest

negative) diagonal element of the matrix. This DTMC is referred to as a uniformized embedded DTMC of the CTMP. Denoting by $\underline{\phi}^{(k)}$ the state probability vector of this DTMC after k transitions, it can be shown (see the above cited references) that

$$\pi_j(t) = \sum_{k=0}^{\infty} \phi_j^{(k)} \frac{(\Lambda t)^k e^{-\Lambda t}}{k!} .$$

For computational purposes, it is necessary to truncate the infinite sum. The truncation error can be easily bounded since we are discarding a Poisson "tail," so that the computational formula becomes

$$\pi_j(t) = \sum_{k=0}^{T(t,\varepsilon)} \phi_j^{(k)} \frac{(\Lambda t)^k e^{-\Lambda t}}{k!} , \tag{3}$$

where

$$T(t,\varepsilon) = \min \left\{ N: \sum_{n=0}^{N} \frac{e^{-\Lambda t}(\Lambda t)^n}{n!} > 1 - \varepsilon \right\} ,$$

ε being the maximum tolerable error (specified by the user). One advantage of this method over numerical integration is an exact bound on the computational error.

The major computational effort in using (3) is now reduced to finding the state probability vector, $\underline{\phi}^{(k)}$, of the uniformized embedded DTMC. This can be readily accomplished by the usual recursion,

$$\underline{\phi}^{(0)} = \underline{\pi}(0) \; ; \quad \underline{\phi}^{(k+1)} = \underline{\phi}^{(k)} P . \tag{4}$$

Gross and Miller (1984a) give a more efficient procedure than the successive vector-matrix multiplication of (4), which takes advantage of the sparsity of the P matrix.

3. Steady-state Environment

Solving for the steady-state probability vector $\underset{\sim}{\pi}$ requires solving the set of linear algebraic equations of (2). Since one of these equations is redundant, it is necessary to reduce the equation set by one and use $1 = \underset{\sim}{\pi}e$ as the final equation. Thus (2) can be reformulated as

$$\underset{\sim}{b} = \underset{\sim}{\pi}A \ , \tag{5}$$

where $\underset{\sim}{b}$ is a vector of all zeroes, except for the last element, which is a 1, and A is the Q matrix with the last column replaced by 1's.

For relatively small systems, the solution can be obtained by inverting A to get

$$\underset{\sim}{\pi} = \underset{\sim}{b}A^{-1} \ .$$

However, for most realistic problems, the state space (and hence dimension of the A matrix) is too large to obtain A^{-1} efficiently or accurately. This situation suggests iterative procedures such as Jacoby or Gauss-Seidel.

Consider the A matrix as a sum,

$$A = L + D + U \ ,$$

where L is a lower triangular matrix, D is a matrix with only diagonal elements, and U is an upper triangular matrix. Then (5) can be written as

$$\underset{\sim}{\pi}(L+D+U) = \underset{\sim}{b}$$

or

$$\underset{\sim}{\pi}D = \underset{\sim}{b} - \underset{\sim}{\pi}(L+U) \ . \tag{6}$$

We can use (6) in an iterative fashion,

$$\underset{\sim}{\pi}^{(n+1)}D = \underset{\sim}{b} - \underset{\sim}{\pi}^{(n)}(L+U) \ , \tag{7}$$

where we begin the procedure with some initial guess, say $\underset{\sim}{\pi}^{(0)}$. This procedure is called Jacoby iteration. Note that in performing the calculations, since D is a diagonal matrix, we compute $\pi_0^{(n+1)}$, $\pi_1^{(n+1)}$, $\pi_2^{(n+1)}$, ... successively. If, as we compute the $\pi_i^{(n+1)}$, we replace the $\pi_i^{(n)}$ on the right-hand side [e.g., in computing $\pi_j^{(n+1)}$, the $\underset{\sim}{\pi}^{(n)}$ vector is modified to be $\underset{\sim}{\pi}^{(n)} = (\pi_0^{(n+1)}, \pi_1^{(n+1)}, \ldots, \pi_{j-1}^{(n+1)}, \pi_j^{(n)}, \pi_{j+1}^{(n)}, \ldots, \pi_N^{(n)})$] , this procedure is referred to as Gauss-Seidel iteration, and in matrix representation is

$$(U^T+D)\underset{\sim}{\pi}^{(n+1)} = \underset{\sim}{b} - L^T\underset{\sim}{\pi}^{(n)} , \tag{8}$$

where $\underset{\sim}{\pi}$ and $\underset{\sim}{b}$ are now column vectors, and U^T, L^T are the transposes of U and L , respectively.

Two questions remain to be answered concerning use of the iterative procedures of (7) or (8); namely, (i) do the procedures converge, and (ii) when should the iterations be terminated? In general, these procedures may not necessarily converge, although for our well-structured Markov process convergence will take place. The stopping criterion generally used is the Cauchy criterion, namely, stop when

$$\max_i \left| \pi_i^{(n+1)} - \pi_i^{(n)} \right| < \varepsilon_0 , \tag{9}$$

where ε_0 is an "arbitrarily" chosen small number. We found using the fractional difference version of (9), namely, stop when

$$\max_i \left| \frac{\pi_i^{(n+1)} - \pi_i^{(n)}}{\pi_i^{(n)}} \right| < \varepsilon_0 , \tag{10}$$

to be somewhat more effective. While there has been some success in using Gauss-Seidel (G-S) on Markov models [see Kaufman, Gopinath, and

Wunderlich (1981)], problems exist with respect to rate of convergence and appropriate stopping criteria. The G-S convergence rate can often be improved by using overrelaxation, that is, by weighting with a coefficient greater than one the $\pi_0^{(n+1)}$, $\pi_1^{(n+1)}$, ..., $\pi_{j-1}^{(n+1)}$ used in calculating $\pi_j^{(n+1)}$ [see Kaufman, *et al.* (1981) or Maron (1982)].

Usually, the G-S procedure is applied to a set of equations with a nonsingular matrix (such as A). Consider a nonsingular matrix M with positive diagonal elements and negative off-diagonal elements. The G-S procedure is known to converge for sets of equations with such an M matrix [see Varga (1963)]. Now consider equation set (2), namely,

$$\underset{\sim}{0} = \underset{\sim}{\pi} Q \ .$$

Multiplying through by -1 gives

$$\underset{\sim}{0} = \underset{\sim}{\pi}[-Q] \ ,$$

where -Q has positive diagonal elements and negative off-diagonal elements. However, it is singular, since one equation of this set is redundant. Suppose we arbitrarily set π_S (assuming there are S states) to one, remove the last row of the Q matrix (call this reduced matrix $\hat{Q}$), and consider solving the reduced S-1 × S-1 set of equations

$$\underset{\sim}{0} = \underset{\sim}{\pi}[-\hat{Q}] \ .$$

Now $-\hat{Q}$ is an M matrix and convergence is guaranteed. Of course the resulting π_i values are relative to $\pi_S = 1$ so that they must be renormalized by dividing each by $\sum_{i=1}^{S} \pi_i$. How fast convergence takes place still is a key question, however. It turns out [see Kaufman, *et al.* (1981)] that working with the full Q matrix, even though it is singular, speeds convergence, and this is what we also do.

Another procedure is to use the uniformized embedded DTMC of the randomization procedure with transition probability matrix $P = Q/\Lambda + I$. This Markov chain has limiting probabilities given by

$$\underset{\sim}{\phi} = \underset{\sim}{\phi}P \ , \tag{11}$$

and they are identical to the $\underset{\sim}{\pi}$ of the CTMP we seek [$\underset{\sim}{\phi} = \underset{\sim}{\phi}P \Rightarrow \underset{\sim}{\phi} = \underset{\sim}{\phi}\left((Q/\Lambda)+I\right) \Rightarrow \underset{\sim}{0} = \underset{\sim}{\phi}(Q/\Lambda) \Rightarrow \underset{\sim}{0} = \underset{\sim}{\phi}Q \equiv \underset{\sim}{0} = \underset{\sim}{\pi}Q$]. Solving the set of equations given by (11) is no easier, of course, than solving that of (5). However, we know from Markov chain theory that limiting probabilities of a DTMC can be found by iteration, namely,

$$\underset{\sim}{\pi}^{(n+1)} = \underset{\sim}{\pi}^{(n)}P \ . \tag{12}$$

Here again, we have computational problems associated with iteration, but we know from Markov chain theory that convergence is guaranteed due to the existence of a steady state vector $\underset{\sim}{\pi}$ (the P matrix is irreducible). The problem of when to stop the iterations remains, however. Using the Cauchy criterion here results in problems similar to those found when using it for G-S iteration, namely, successive probabilities can differ by very small amounts and still be far from the steady state values.

Wallace and Rosenberg (1966) provide a considerably better stopping criterion than the Cauchy criterion of (10). Their stopping rule is based on estimating the rate of convergence by estimating the second eigenvalue of P , and turns out to be: "Stop when

$$\frac{\left|\pi^{(n+1)} - \pi^{(n)}\right|}{\left(1 - \left|\pi^{(n+1)} - \pi^{(n)}\right|\right)^{1/n}} < \varepsilon_0 \ ." \tag{13}$$

For details of this development, see Wallace and Rosenberg (1966) or Gross, Kioussis, Miller, and Soland (1984).

4. Results

The following section gives a brief summary of results to date. For greater detail, we refer the reader to Gross and Miller (1984b) and Gross, Kioussis, and Miller (1984) for the transient case and to Gross, Kioussis, Miller, and Soland (1984) for the steady-state case.

4.1 Transient Case

The largest system solved to date using equation (3) directly was a (2,2,2) system (as pictured in Figure 1) with 18 components at base 1 (of which 4 were spares), 13 at base 2 (of which 3 were spares), and 3 spares at the depot. The base repair shops had 2 parallel service channels each, and the depot repair facility had four. This gave a state space of 20,748 ($Q = 20{,}748 \times 20{,}748$).

The time-varying environment scenario is shown in Figure 2. At time 6, a shift in MTTF ($1/\lambda$) occurs but it takes until time 10 for the repair facilities to "catch up" in MTTR ($1/\mu$). This simulates a change in usage due to, say, a shift from peacetime to wartime. The measure of effectiveness calculated is the availability at time t ($t = 1,2,\ldots,15$), where availability is defined as follows:

$A_1(t) \equiv$ Pr{at least 14 components are operational at base 1 at time t}

$A_2(t) \equiv$ Pr{at least 10 components are operational at base 2 at time t}

$A_{12}(t) \equiv$ Pr{at least 14 components at base 1 and at least 10 components at base 2 are simultaneously operational at time t}.

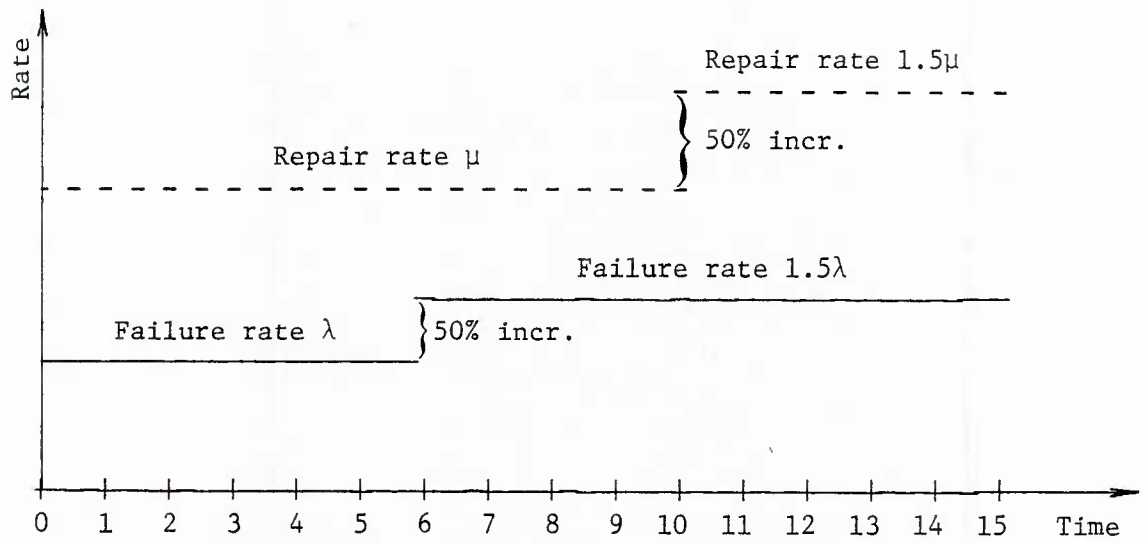


Figure 2. Time-varying environment scenario for sample run.

Figure 3 shows a plot of $A_1(t)$ versus t. Plots of $A_2(t)$ and $A_3(t)$ are similar in nature. The graph shows an initial $A_1(0)$ of 1.0 (we assume at time zero *all* components are operational) and thereafter a drop-off toward the steady-state availability as time increases. At time 6, the increase in failure rate occurs and $A_1(t)$ begins to drop off at a higher rate, heading for a new, lower steady-state availability. However, the increase in repair rate at time 10 causes $A_1(t)$ to begin to rise, heading back toward the original steady-state availability. This run took approximately 25 minutes of CPU time on a VAX 11/780 computer using the randomization computation of (3) with the efficient procedure given in Gross and Miller (1984a) for calculating $\underline{\phi}^{(k)}$.

As the systems become more complex (more bases, multiple component types, indenture, more echelons, etc.) the state-space grows rapidly. We have solved a problem with three bases, yielding a state-space of size 43,278,703, by truncating the state-space ("lumping" low probability states into several absorbing states resulting in a truncated state space

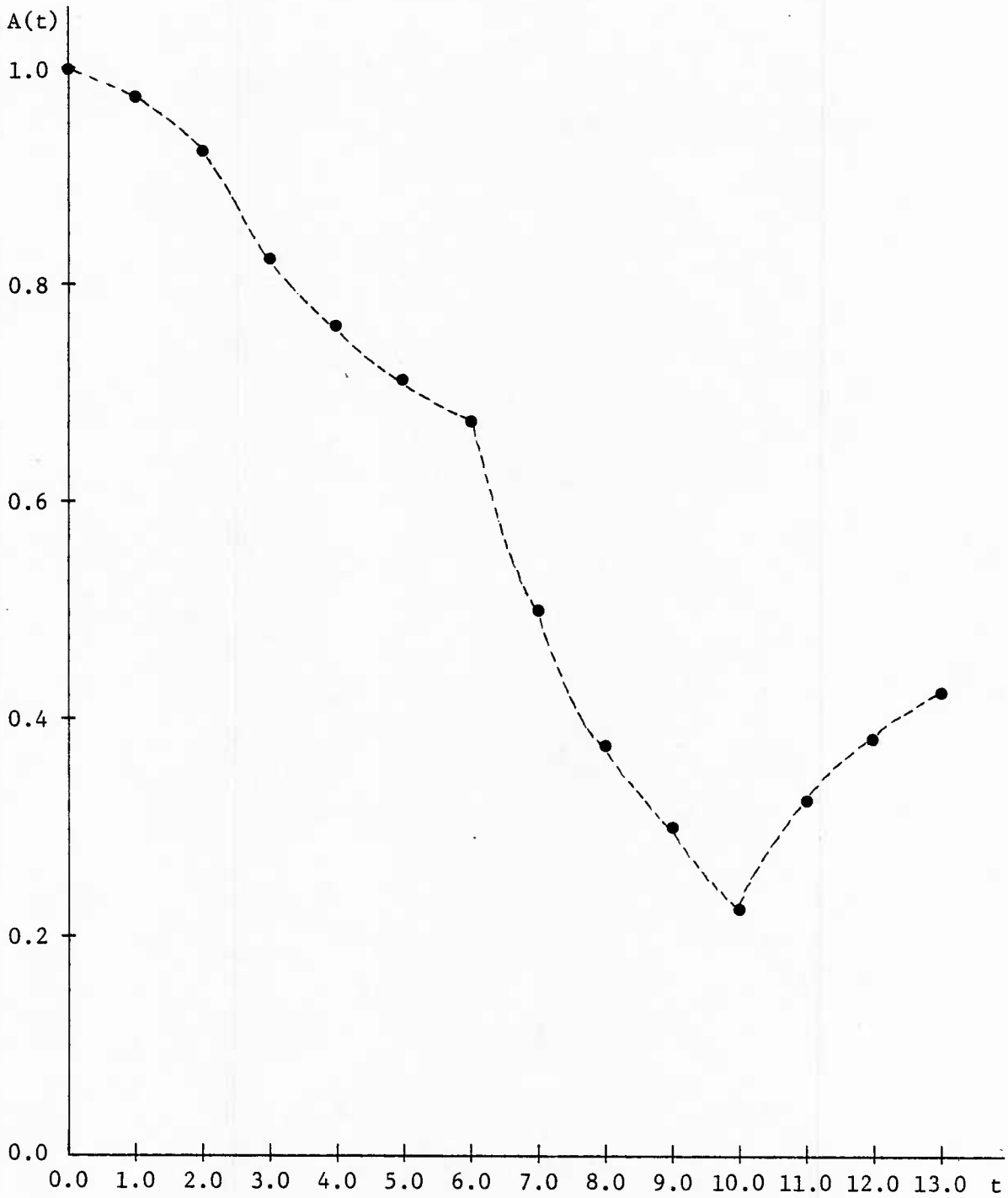


Figure 3. $A_1(t)$ versus t for sample run.

of approximately 15,000 states) on the VAX 11/780 in approximately 30 minutes [see Gross, Kioussis, and Miller (1984)].

4.2 Steady-state Case

Ironically, computational success has been far more elusive for the steady-state situation than for the transient case. The problem is the stopping criterion for these iterative procedures (a problem not present when dealing with transient solutions). In the transient case, the randomization procedure guarantees an accuracy to within a pre-specified ε. For steady state, using either the Cauchy or the Wallace-Rosenberg stopping rule does not guarantee errors within ε. Table 1 shows some computations for a (1,1,1) system which is the classical machine repair with spares model of queueing theory. For this model the availability can be computed analytically, which allowed us to estimate the actual error. The columns under P-WR show the results of using (12) with the stopping criterion of (13), the Wallace-Rosenberg approach, while the GS-C columns show results for (8) with the stopping criterion of (10), the Gauss-Seidel approach.

The circled elements show the cases for which the error specification, ε, was exceeded. While there were more cases of exceeding the stopping rule error specification in P-WR, the error excesses were larger, especially for the larger population cases, under GS-C. But GS-C stopped in far fewer iterations in almost all cases (except for the very small population cases), and it is the number of iterations that consumes most of the CPU time.

The last column shows a rerun of GS-C, ignoring the stopping criterion and performing the same number of iterations as used for the P-WR procedure. The errors essentially went to zero, which indicates that if a better stopping criterion could be found, Gauss-Seidel iteration

TABLE 1

RESULTS FOR (1,1,1) MODEL

Desired # Operating (M)	# Spares (Y)	# Repair Channels (C)	$\rho \equiv M\lambda/C\mu$	Exact Avail*	P-WR		GS-C		GS-WR Error
					Error in Avail.	# Iter	Error in Avail.	# Iter	
5	1	1	1.5	.2123	.0071	8	.0019	11	
			1.0	.4434	.0123	19	.0023	12	
			0.5	.7951	.0044	20	.0005	12	
10	5	3	1.5	.1533	.0137	36	.0030	20	
			1.0	.5719	.0122	64	.0136	30	.0001
			0.5	.9745	.0017	55	.0004	32	
20	4	5	1.5	.0386	.0024	57	.0032	42	
			1.0	.3217	.0113	98	.0072	52	.0000
			0.5	.8790	.0034	71	.0004	46	
50	5	3	1.5	.0039	.0003	183	.0006	94	
			1.0	.3640	.0146	313	.0157	148	.0000
			0.5	.9714	.0017	156	.0000	117	
100	10	3	1.5	.0001	.0000	467	.0000	207	
			1.0	.4475	.0152	789	.0385	317	.0000
			0.5	.9991	.0003	302	.0000	241	
100	20	5	1.0	.6025	.0125	1100	.0954	321	.0001

*Percentage of time the population is operating at full strength M .

might be a viable approach. Runs for some (2,2,2) systems and more detailed discussion of these steady-state procedures can be found in Gross, Kioussis, Miller, and Soland (1984).

References

Arsham, H., A. R. Balana, and D. Gross (1983). Numerical solutions of machine repair problems, *Computers & Industrial Engineering*, *7*, 149-157.

Grassmann, W. (1977a). Transient solutions in Markovian queueing systems, *Computers & Operations Research*, *4*, 47-56.

Grassmann, W. (1977b). Transient solutions in Markovian queues, *European Journal of Operational Research*, *1*, 396-492.

Gross, D. and D. R. Miller (1984a). The randomization technique as a modelling tool and solution procedure for transient Markov processes, *Operations Research*, *32*, 343-361.

Gross, D. and D. R. Miller (1984b). Multi-echelon repairable item provisioning in a time-varying environment using the randomization technique, *Naval Research Logistics Quarterly*, in press.

Gross, D., L. C. Kioussis, and D. R. Miller (1984). Transient behavior of large Markovian multi-echelon repairable item inventory systems using a truncated state-space approach, Technical Paper GWU/IMSE/Serial T-488/84, Institute for Management Science and Engineering, George Washington University

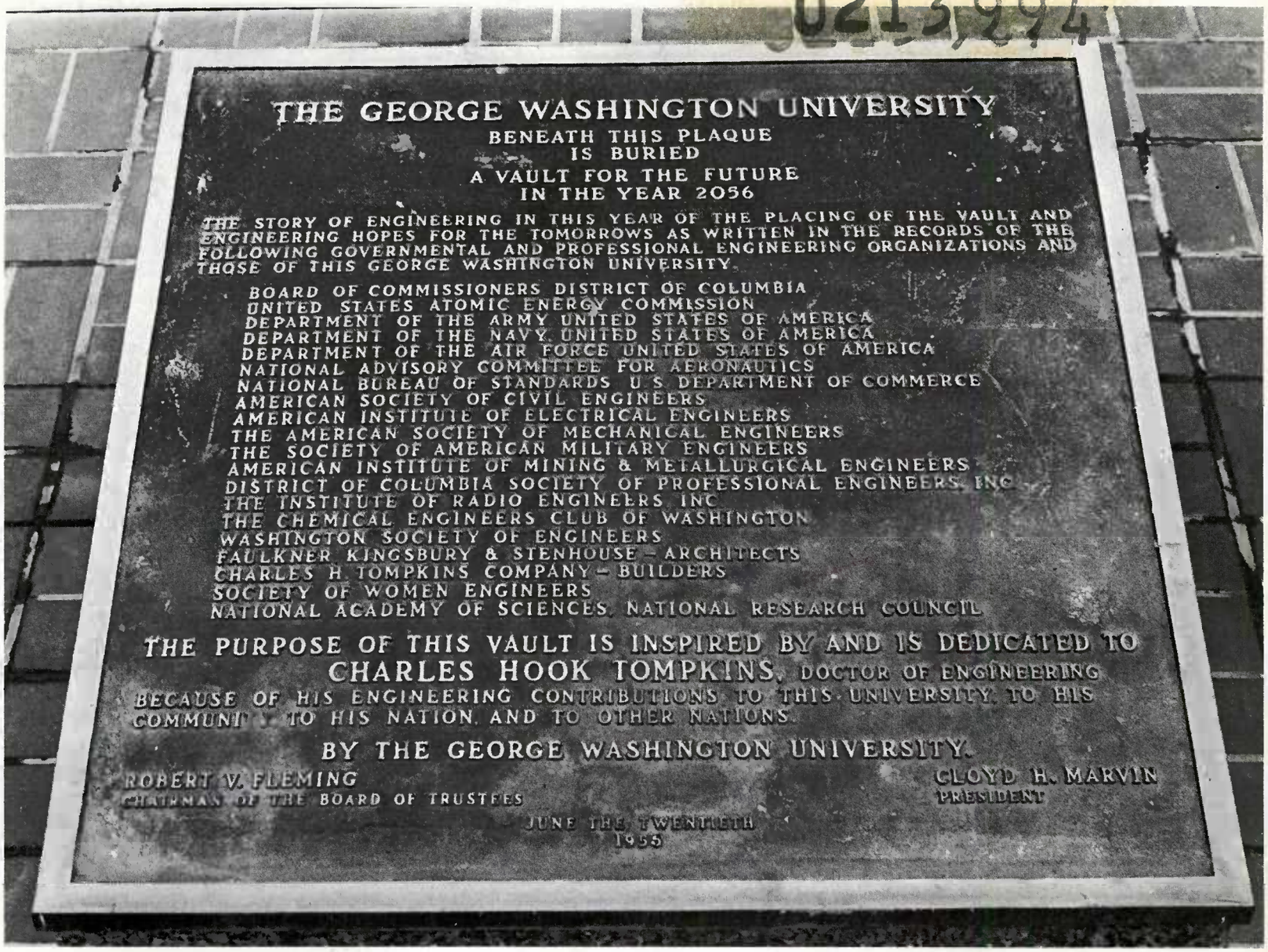
Gross, D., L. C. Kioussis, D. R. Miller, and R. M. Soland (1984). Computational aspects for determining steady state availability in Markovian multi-echelon repairable item inventory models, Technical Paper GWU/IMSE/Serial T-483/84, Institute for Management Science and Engineering, George Washington University.

Kaufman, L., G. Gopinath, and E. F. Wunderlich (1981). Analysis of packet network congestion control using sparse matrix algorithms, *IEEE Transactions on Communications*, *COM-29*, 453-464.

Maron, M. J. (1982). *Numerical Analysis*. Macmillan, New York.

Varga, R. S. (1963). *Matrix Iterative Analysis*. Prentice-Hall, Englewood Cliffs, New Jersey.

Wallace, V. L. and R. S. Rosenberg (1966). RQA-1, the recursive queue analyzer, Systems Engineering Laboratory Report 97742-1-T, Department of Electrical Engineering, University of Michigan, Ann Arbor.



To cope with the expanding technology, our society must be assured of a continuing supply of rigorously trained and educated engineers. The School of Engineering and Applied Science is completely committed to this objective.